SCHWEIZERISCHE EIDGENOSSENSCHAFT
CONFÉDÉRATION SUISSE
CONFEDERAZIONE SVIZZERA

Bescheinigung

Die beiliegenden Akten stimmen mit den ursprünglichen technischen Unterlagen des auf der nächsten Seite bezeichneten Patentgesuches für die Schweiz und Liechtenstein überein. Die Schweiz und das Fürstentum Liechtenstein bilden ein einheitliches Schutzgebiet. Der Schutz kann deshalb nur für beide Länder gemeinsam beantragt werden.

Attestation

Les documents ci-joints sont conformes aux pièces techniques originales de la demande de brevet pour la Suisse et le Liechtenstein spécifiée à la page suivante. La Suisse et la Principauté de Liechtenstein constituent un territoire unitaire de protection. La protection ne peut donc être revendiquée que pour l'ensemble des deux Etats.

Attestazione

I documenti allegati sono conformi agli atti tecnici originali della domanda di brevetto per la Svizzera e il Liechtenstein specificata nella pagina seguente. La Svizzera e il Principato di Liechtenstein formano un unico territorio di protezione. La protezione può dunque essere rivendicata solamente per l'insieme dei due Stati.

Bern, 30. JAN. 2003

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Eidgenössisches Institut für Geistiges Eigentum
Institut Fédéral de la Propriété Intellectuelle
Istituto Federale della Proprietà Intellettuale

Patentverfahren
Administration des brevets
Amministrazione dei brevetti

Rolf Hofstetter

Demande de brevet no 2002 0294/02

CERTIFICAT DE DEPOT (art. 46 al. 5 OBI)

L'Institut Fédéral de la Propriété Intellectuelle accuse réception de la demande de brevet Suisse dont le détail figure ci-dessous.

Titre:
Méthode de stockage de blocs de données dans une mémoire.

Requérant:
NagraCard S.A.
22, route de Genève
1033 Cheseaux-sur-Lausanne

Mandataire:
Leman Consulting S.A.
62 rte de Clementy
1260 Nyon

Date du dépôt: 20.02.2002

Classement provisoire: B65G

METHODE DE STOCKAGE DE BLOCS DE DONNEES DANS UNE MEMOIRE

La présente invention est du domaine du stockage de données dans une mémoire numérique réinscriptible à semi-conducteurs qui conserve son contenu en cas de coupure de courant. Plus particulièrement, l'invention concerne la gestion de la place mémoire disponible par une méthode de stockage de blocs de données dans la mémoire.

Les mémoires à semi-conducteurs sont utilisées dans toutes les applications comprenant des microprocesseurs pour lesquelles il est nécessaire de stocker le programme et les données nécessaires à leur fonctionnement.

Les données sont en général introduites dans la mémoire à des adresses prédéterminées c'est-à-dire définie lors du l'élaboration du programme, ou soit de manière séquentielle c'est-à-dire par blocs successifs à la suite des données déjà présentes dans la mémoire. Ces blocs peuvent également être réinscrits à la place d'autres blocs présents dans le but de renouveler des données devenues obsolètes. Un bloc est une suite de bit ou d'octets de longueur ou de taille prédéfinie comportant une entête contenant un identificateur du bloc et un nombre définissant sa longueur.

Suivant les instructions du programme, les données sont stockées dans la mémoire à des emplacements définis par des adresses. Ces dernières sont fixées par des paramètres contenus dans le programme. Ces emplacements réservés sont situés dans une zone quelconque de la mémoire dont les limites sont définies par une plage d'adresses. Cet intervalle ainsi déterminé correspond à la capacité disponible qui est en général plus grande que la quantité maximale de données pouvant y être stockée.

De nombreuses applications de traitement de données de plus en plus sophistiquées sont embarquées sur des supports physiques de taille de plus en plus réduite. Par conséquent, la capacité des mémoires utilisées par les microprocesseurs devra être optimisée au maximum. Ces cas se présentent par exemple dans différents modules électroniques comme les cartes à puces ou tout autre support comportant des composants de traitement numérique de données miniaturisés.

Certaines applications notamment de contrôle d'accès, d'identification d'un utilisateur ou de paiement électronique, doivent répondre à des exigences de sécurité de plus en plus élevées afin d'éviter les fraudes. En effet, les fonctionnalités d'une carte peuvent être révélées suite à des analyses approfondies du contenu de la mémoire associée au processeur. Par exemple, le mécanisme d'un débit sur une carte de paiement produit un ensemble de données qui vont être stockées à des emplacements de la mémoire prédéterminés par le programme. A chaque opération effectuée par la carte correspond une configuration bien définie des données dans la mémoire. Cette situation laisse une porte ouverte au piratage des cartes dont les fonctionnalités peuvent être copiées ou simulées sur d'autres cartes.

Le but de la présente invention est de proposer une méthode de stockage sécurisé de données dans une mémoire de manière à éviter les contrefaçons par analyse de son contenu. Un autre but consiste à limiter l'usure de la mémoire par une gestion améliorée des zones de lecture / écriture des données.

Ce but est atteint par une méthode de stockage d'une pluralité de blocs de données dans une mémoire numérique à semi-conducteurs réinscriptible pilotée par un gestionnaire de mémoire caractérisée par les étapes suivantes:

- déterminer aléatoirement une zone disponible,
- stocker le bloc de données dans la zone ainsi choisie.

On entend par zone disponible une zone de la mémoire libre de données ou qui contient des données remplaçables par de nouvelles comme dans le cas d'une mise à jour par exemple.

La méthode selon l'invention permet le stockage de blocs de données dans des emplacements de mémoire toujours différents même si le programme effectue une suite d'opérations identiques. Par exemple une opération de débit de 10 unités sur une carte n'aura pas le même effet sur la structure du contenu de la mémoire à chaque exécution du même débit. De plus deux cartes identiques qui exécutent une opération identique auront une structure du contenu de leur mémoire complètement différente. Ainsi une analyse des données d'une carte ne permet pas de reproduire une image des opérations de la première carte sur l'autre et vice versa.

En plus de l'aspect sécurité, la méthode de l'invention permet, grâce à la lecture / écriture dans des zones choisies aléatoirement, une meilleure répartition de l'usure de la mémoire. Il n'y aura donc pas de régions dans la mémoire qui s'usent plus rapidement que d'autres comme dans le cas où de nombreux cycles de lecture / écriture de données s'effectuent toujours à un endroit réservé dans la mémoire.

Le choix aléatoire d'une zone mémoire disponible peut s'effectuer selon plusieurs variantes:

1- Le résultat obtenu après l'exploration de la mémoire constitue une liste d'adresses correspondant à des zones disponibles. Cette liste est conservée temporairement dans une mémoire vive. Une adresse est ensuite choisie aléatoirement dans cette liste, puis le bloc de données est stocké dans la zone de la mémoire indiquée par cette adresse. Une variante de cette méthode consiste à maintenir en permanence une table des zones disponibles et de choisir une adresse aléatoirement parmi celles-ci.

2- L'exploration de la mémoire détermine le nombre maximum de zones disponibles. Un choix aléatoire d'un nombre n compris entre 1 et au le nombre de zones trouvées désigne la zone où le bloc doit être stocké. Par exemple 20 zones disponibles sont trouvées, le choix aléatoire d'un nombre compris entre 1 et 20 donne 8, le bloc est alors stocké dans la $8^{ème}$ zone disponible.

3- Un nombre N est déterminé aléatoirement entre 1 et le nombre maximum de zones possibles. Le gestionnaire de mémoire cherche séquentiellement la $N^{ème}$ zone disponible et si la fin de la mémoire est atteinte avant que cette zone ne soit trouvée, le gestionnaire reprend la recherche depuis le début de la mémoire jusqu'à ce que la $N^{ème}$ zone disponible soit trouvée.

L'invention sera mieux comprise grâce à la description détaillée qui va suivre et qui se réfère aux dessins annexés qui sont donnés à titre d'exemple nullement limitatif, à savoir:

La figure 1 montre le stockage de blocs de données de longueur égale dans une portion de mémoire.

La figure 2 montre le stockage de blocs de longueur variable.

La figure 3 illustre le stockage de blocs en tenant compte d'un pas prédéterminé.

La figure 1 illustre un cas où les blocs de données ont tous une longueur l égale. Ils sont mémorisés aléatoirement dans des zones disponibles dont la longueur correspond à un multiple de la longueur d'un bloc à mémoriser. Par exemple les blocs ont tous une longueur de 10 octets, ils pourront être répartis au hasard dans des emplacements de 10, 20, 30, 40, etc. octets. La zone disponible peut être plus grande que le bloc à mémoriser. Par exemple un bloc de 10 octets B8 peut être placé dans un espace e2 de 30 octets et avec un décalage de 20 octets par rapport au début de l'emplacement disponible, c'est-à-dire à 20 octets du bloc précédent B5.

Lors du stockage d'un nouveau bloc Bn, selon la première variante de l'invention, le gestionnaire de mémoire va explorer la mémoire et en déduire les adresses disponibles de e1, e2,1, e2,2, e3 et e4 étant entendu que l'espace e2 permet de stocker deux blocs de longueur fixe. Une fois ces adresses déterminées, une variable aléatoire peut être utilisée pour définir l'adresse de la zone disponible où sera stocké le bloc Bn.

Selon la seconde variante, le gestionnaire trouve 5 zones disponibles dont la longueur correspond à celle des blocs à stocker. Un choix au hasard d'un nombre compris entre 1 et 5 donne 3, le bloc Bn sera donc stocké dans la 3ème zone, c'est-à-dire dans e2,2.

Selon la troisième variante, le nombre de maximum de zones disponibles Z est 13. Le gestionnaire détermine aléatoirement un nombre N entre 1 et 13, par exemple 8 puis il parcourt la mémoire pour trouver la 8ème place disponible. Une première passe permet de révéler que 5 zones sont disponibles et une seconde passe depuis le début détermine que l'emplacement e2,2 (le 3ème) correspond à la 8ème place. En résumé, si le nombre aléatoire N déterminé est plus grand que le nombre de places disponibles P, le rang de l'espace libre est défini par le nombre aléatoire N modulo le nombre de places P disponibles. Ici dans l'exemple, N=8 étant plus grand que P=5, alors le bloc sera stocké à la place 8 modulo 5 = 3ème place. Dans le cas particulier où N modulo P donne 0, le bloc peut être placé à la première ou à la dernière place. Selon une autre variante, le nombre aléatoire N peut être redéfini jusqu'à obtenir une valeur N modulo P différente de zéro.

La figure 2 a) représente le cas où les blocs ont une longueur variable et sont séparés ou non par des zones libres. Par exemple un bloc B2 de 20 octets débute à 5 octets du bloc précédent et se termine 5 octets avant le bloc B4. Les zones ou espaces libres e1 et e2 autour de B2 pourront être occupés si B2 et B4 par exemple doivent être remplacés. Il en va de même pour tous les autres espaces libres qui sont soit occupés, soit se déplacent lors du stockage de nouveaux blocs Bn à la place des précédents.

Un nouveau bloc Bn pourra être stocké dans les espaces libres restants ou se substituer à un ou plusieurs blocs déjà présents et devenus inutiles. L'espace ainsi libéré permet le stockage de plusieurs blocs plus petits ou d'un bloc plus grand occupant tout ou partie de l'espace. Les figures b) et c) montrent un exemple de mise à jour: un nouveau bloc B12 a été stocké dans l'espace libre e4. B10 est remplacé par un bloc B11 plus grand occupant ainsi tout l'espace e9 libéré entre B7 et B9. Les blocs B2 et B4 ont été remplacés par B13 occupant la moitié de l'espace e10 libéré. Le nouvel espace libre e11 ainsi créé sera exploité lors d'un prochain stockage de blocs.

Selon une autre variante de la méthode de l'invention illustrée par la figure 3, le programme détermine une longueur courante m des blocs de données à mémoriser. Cette valeur peut correspondre à la longueur la plus fréquente des blocs ou dans certains cas à la longueur moyenne des blocs. Après le choix aléatoire de la zone de stockage disponible, le bloc sera mémorisé soit directement à la suite d'un bloc déjà présent dans le cas où ce bloc est de longueur égale ou supérieure à cette longueur m, soit avec un décalage de n octets afin que la longueur du bloc et du décalage n soit égale à la longueur m. Cette variante permet, lors de l'effacement de ce bloc, de libérer un espace qui sera très rapidement utilisé. Sans ce décalage prévu lors du stockage, la place libérée par ce bloc aura très peu de chance d'être réutilisée.

Selon notre exemple la longueur courante m des blocs est de 15 octets, les blocs ont des longueurs variant entre 5 et 20 octets. Deux cas se présentent:

Si la longueur du bloc Bn à stocker est plus petite que la longueur m courante, Bn est stocké à un pas m du bloc précédent de façon à laisser un espace libre égal à la différence entre m et la longueur de Bn. Selon l'exemple ci-dessus, un bloc de 10

octets se place à 15-10 = 5 octets du bloc précédent. La figure 3 a) montre des blocs séparés par des zones disponibles. Dans la figure 3 b), un bloc B6 est stocké dans l'espace libre e2, la longueur de B6 étant plus petite que la longueur courante m, B6 se place à un pas m à partir du bloc précédent B2. L'espace e5 entre B2 et B6 équivaut à la différence de longueur entre m et la longueur de B6.

Si la longueur du bloc Bn à stocker est plus grande ou égale à la longueur m courante, Bn se place immédiatement après le bloc précédent. Dans la figure 3 b), B7 est plus grand que la valeur m, il se place donc dans e4 à la suite de B5 sans laisser d'espace libre entre eux.

La méthode selon l'invention peut aussi s'appliquer à des mémoires plus importantes ayant une structure en forme de table ou de matrice permettant un accès direct aux blocs de données. Des pointeurs définissent dans ce cas les emplacements disponibles dans la mémoire. Ces derniers sont choisis aléatoirement avant le stockage des blocs de données dans la mémoire.

Les données dont les blocs ont été stockés selon la méthode de l'invention peuvent être reconstituées par l'analyse, soit des identificateurs contenus dans les entêtes des blocs, soit des adresses de chaque bloc contenues dans une table préalablement mémorisée.

REVENDICATIONS

1. Méthode de stockage d'une pluralité de blocs de données dans une mémoire numérique à semi-conducteurs réinscriptible pilotée par un gestionnaire de mémoire caractérisée par les étapes suivantes:

- déterminer aléatoirement une zone disponible,

- stocker le bloc de données dans la zone ainsi choisie.

2. Méthode selon la revendication 1 caractérisée en ce qu'elle comprend une étape préalable d'exploration de la mémoire effectuée par le gestionnaire, ladite exploration déterminant les zones disponibles.

3. Méthode selon la revendication 2 caractérisée en ce que le résultat obtenu après l'exploration de la mémoire constitue une liste d'adresses de zones disponibles, conservée temporairement dans une mémoire, une adresse est ensuite choisie aléatoirement dans ladite liste, puis le bloc de données est stocké dans la zone de la mémoire indiquée par cette adresse.

4. Méthode selon les revendication 2 caractérisée en ce que l'exploration de la mémoire détermine le nombre de zones disponibles, un choix aléatoire d'un nombre compris entre 1 et le nombre de zones trouvé désigne la zone où le bloc doit être stocké.

5. Méthode selon le revendication 1 caractérisée en ce qu'un nombre N est déterminé aléatoirement compris 1 et le nombre maximum de zones disponibles possibles, le gestionnaire de mémoire cherche séquentiellement la $N^{ème}$ zone disponible et si la fin de la mémoire est atteinte avant que ladite zone ne soit trouvée, le gestionnaire reprend la recherche depuis le début de la mémoire jusqu'à ce que la $N^{ème}$ zone disponible soit trouvée.

6. Méthode selon les revendications 1 à 5 caractérisée en ce que les blocs sont de longueur variable, le stockage d'un bloc dans la mémoire s'effectue dans une zone disponible de longueur égale ou supérieure à la longueur du bloc.

7. Méthode selon les revendications 1 à 5 caractérisée en ce que les blocs de données sont tous de même longueur, les zones de mémoire disponibles ayant une longueur égale ou supérieure à un multiple de la longueur des blocs.

8. Méthode selon les revendications 1 à 6 caractérisée en ce qu'elle comporte une étape préalable de détermination de la longueur courante m des blocs à mémoriser, les blocs Bn de longueur plus petite que ladite valeur courante sont stockés à un pas m du bloc précédent de façon à laisser un espace libre égal à la différence entre la longueur courante (m) et la longueur du bloc (Bn), les blocs (Bn) de longueur égale ou plus grande que la longueur courante (m) sont stockés immédiatement après le bloc précédent.

9. Méthode selon la revendication 1 caractérisée en ce que la mémoire a une structure en forme de table permettant un accès direct aux données au moyen de pointeurs, lesdits pointeurs sont choisis aléatoirement avant le stockage des blocs de données dans la mémoire.

ABREGE

La présente invention décrit une méthode de stockage d'une pluralité de blocs de données dans une mémoire numérique à semi-conducteurs réinscriptible pilotée par un gestionnaire de mémoire caractérisée par les étapes suivantes:

- déterminer aléatoirement une zone disponible,

- stocker le bloc de données dans la zone ainsi choisie.

Cette méthode de stockage de données s'applique de préférence aux cartes à puce et à des modules électroniques similaires. Elle empêche la reproduction des fonctionnalités de la carte suite à une analyse du contenu de la mémoire. De plus, elle assure une meilleure répartition de l'usure de la mémoire.

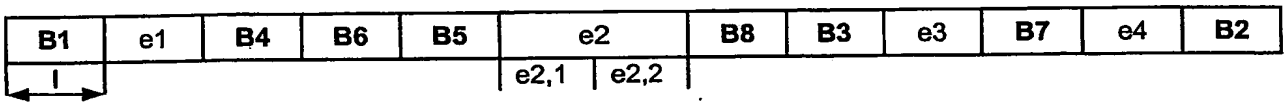

Fig. 1

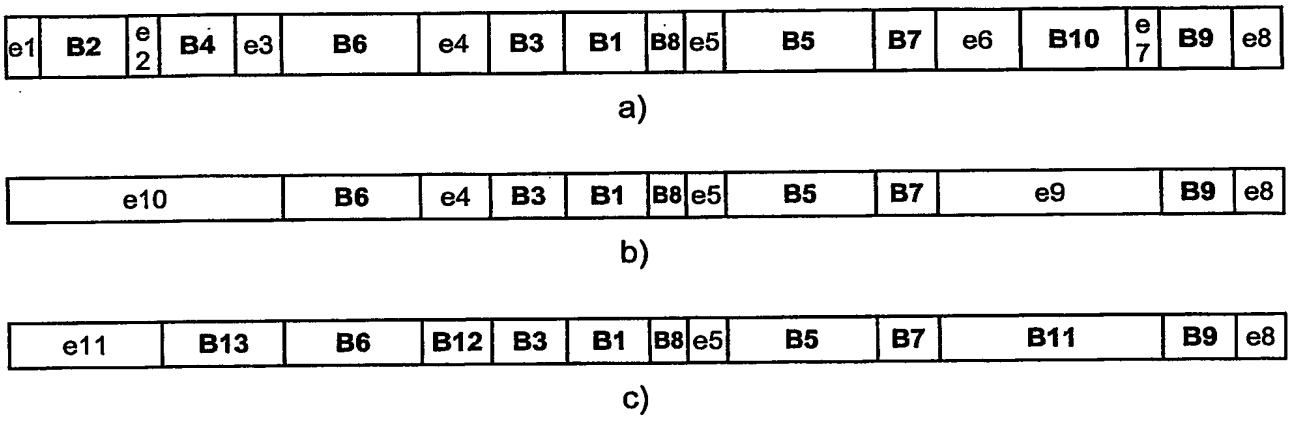

Fig.2

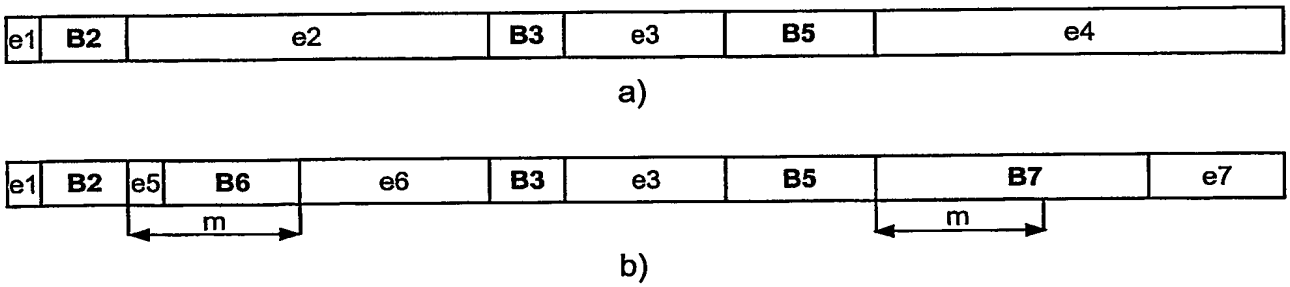

Fig. 3